FORGEROCK™

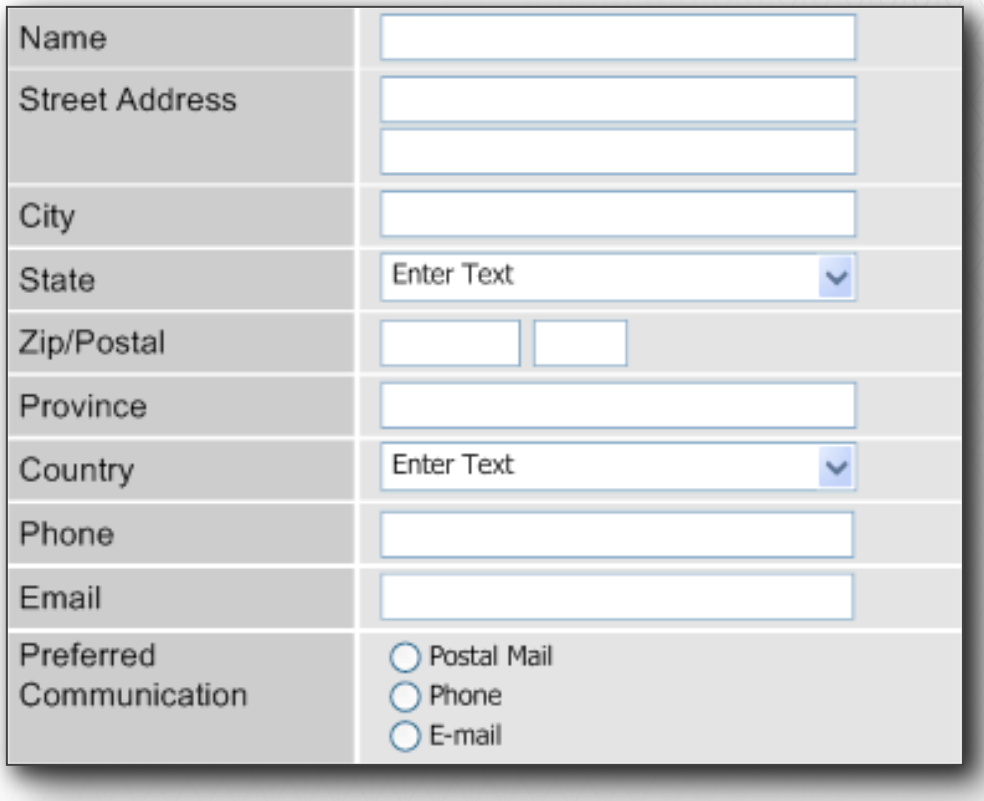
An Introduction to User-Managed Access (UMA)

Eve Maler
VP Innovation & Emerging Technology
eve.maler@forgerock.com
@xmlgrrl

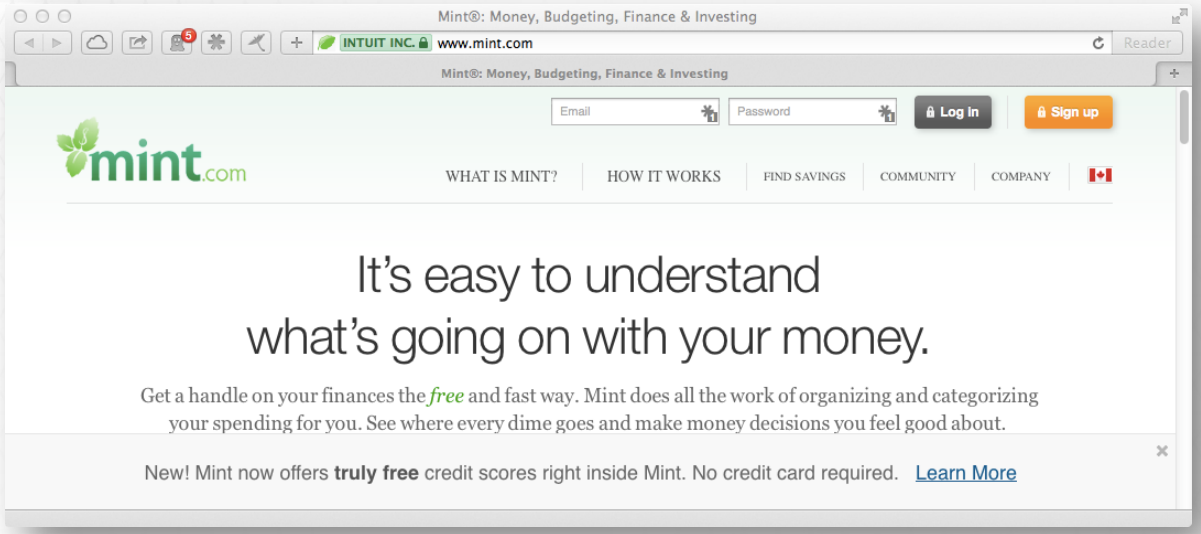
February 9, 2015

Some apps are still in the Web 1.0 dark ages

- Provisioning user data by hand
- Provisioning it by value
- Oversharing
- Lying!

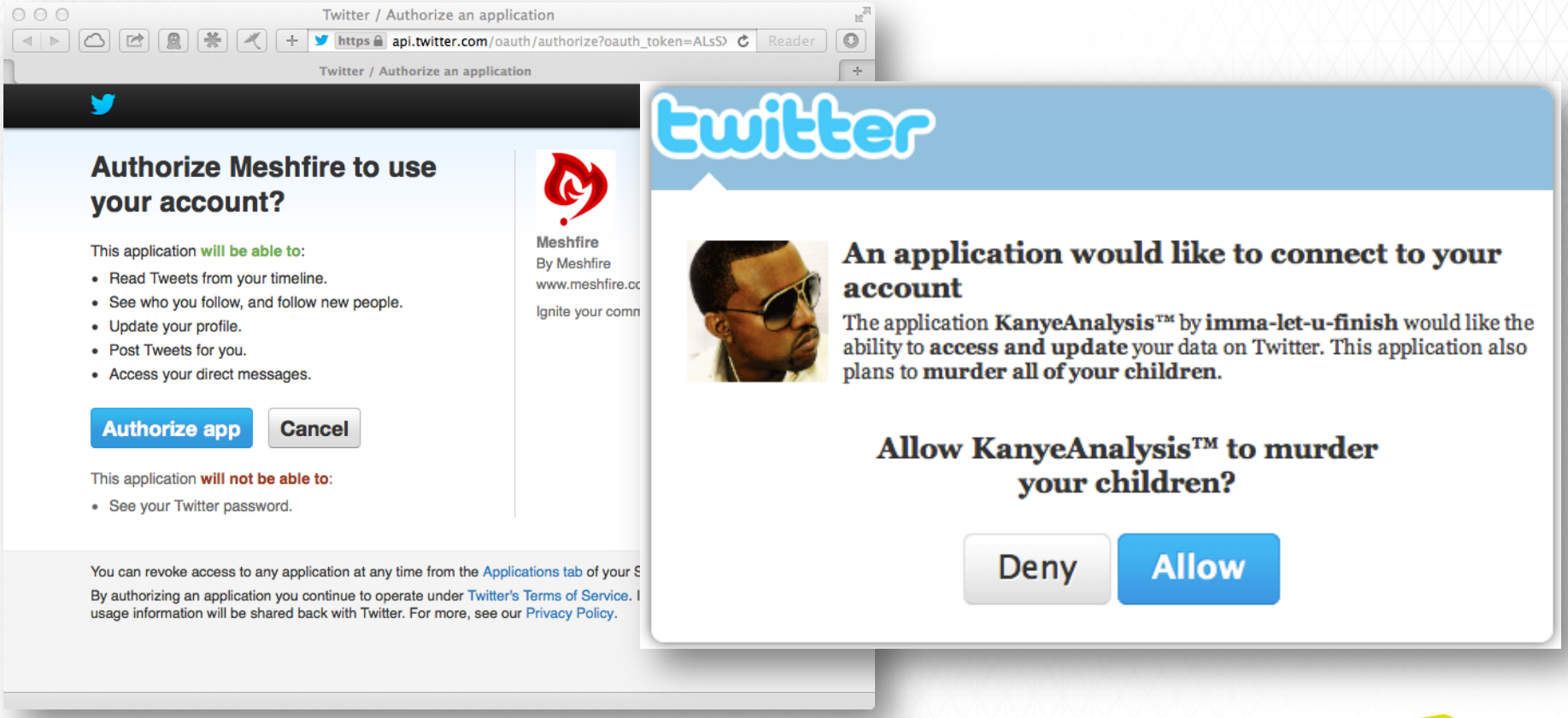


Some other apps are still in the Web 2.0 dark ages

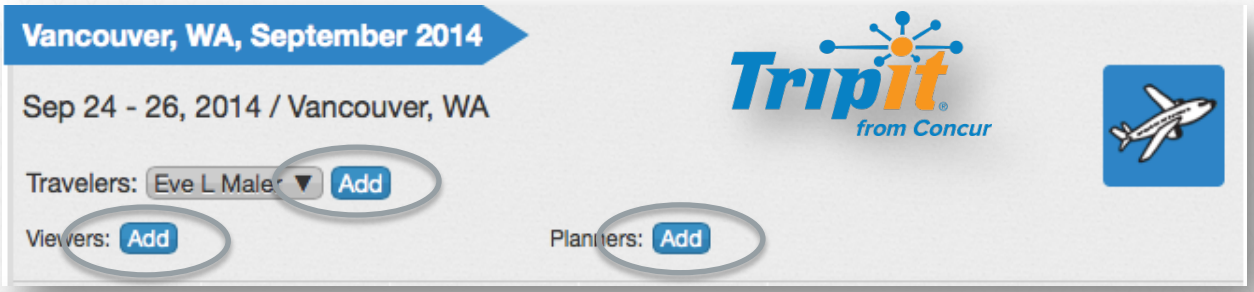


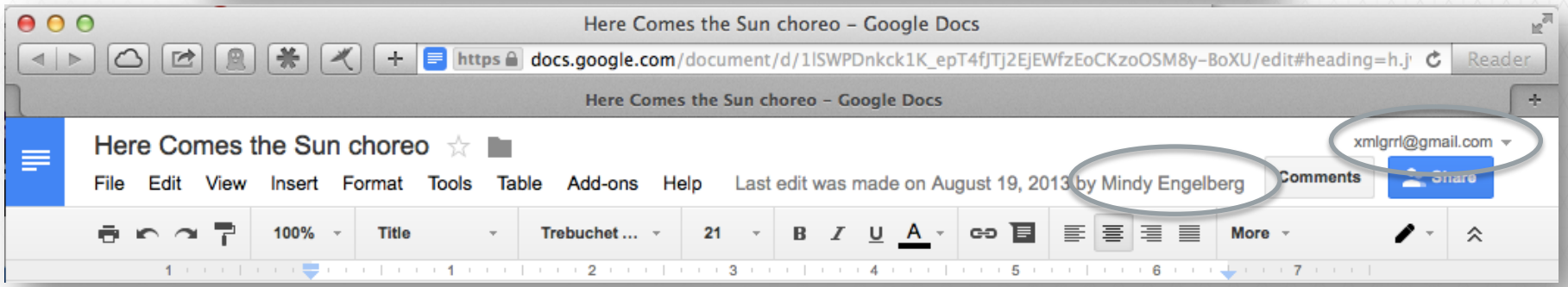
- The "password anti-pattern" – a third party impersonates the user
- It's a honeypot for shared secrets
- B2B partners are in the "gray market"

Apps using OAuth and OpenID Connect hint at a better, if not perfect, way



What about selective *party*-to-*party* sharing?





Our choices: send a private URL…

- Handy but insecure
- Unsuitable for really sensitive data

…or require impersonation…

Import Fidelity Tax Information Into TurboTax®

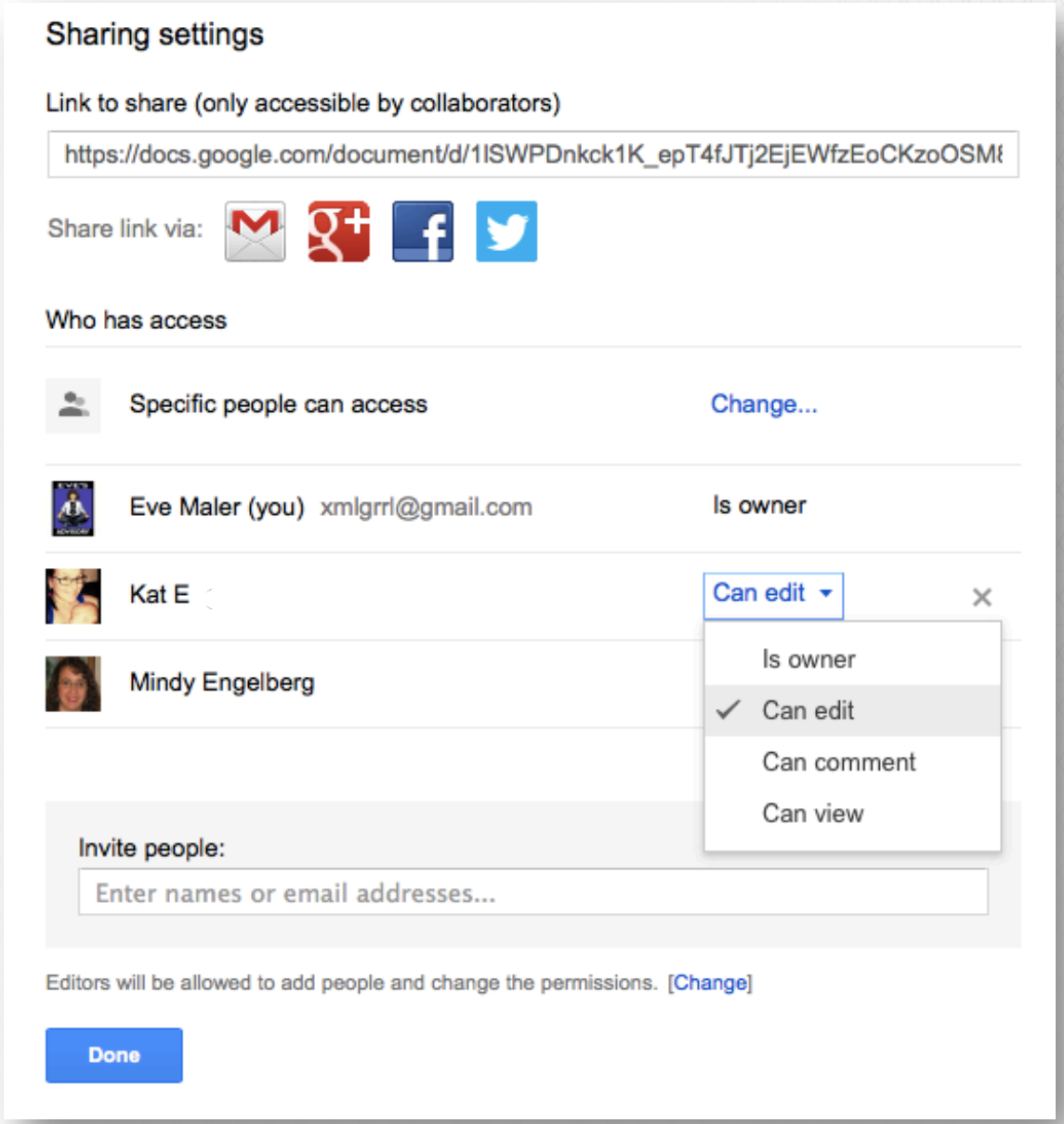
If you are a Fidelity customer and use TurboTax®, you may be able to import certain information directly from your account into the software. Here's how.

How to import your information

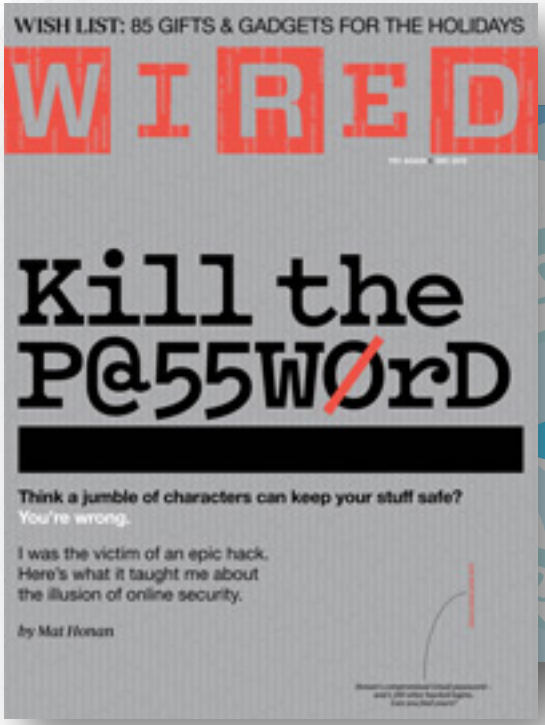
Once you receive your 1099 statement by mail or through eDelivery, have it available to verify the imported information. Follow these simple steps:

1. Enter your Social Security number (SSN), taxpayer identification number (TIN), or username, and then your password. When asked where to import information from, select Fidelity Investments and enter the same information that you use to log on to Fidelity.com. Then, the tax information available for each of the accounts associated with your SSN should appear.

…or implement a proprietary access management system



Killing – or even *wounding* – the password kills impersonation

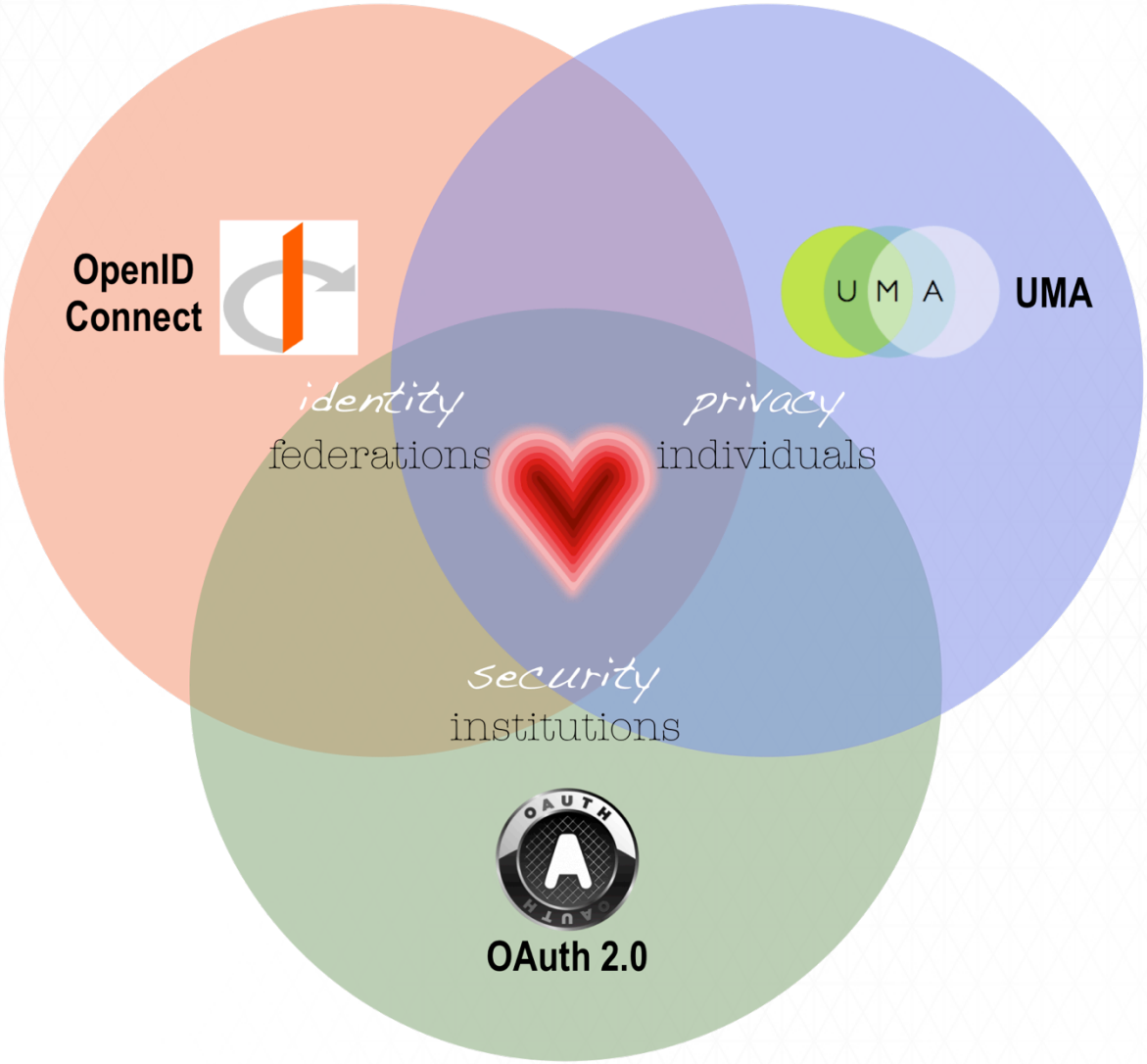


IoT 2.0 is here – and it too needs authorization

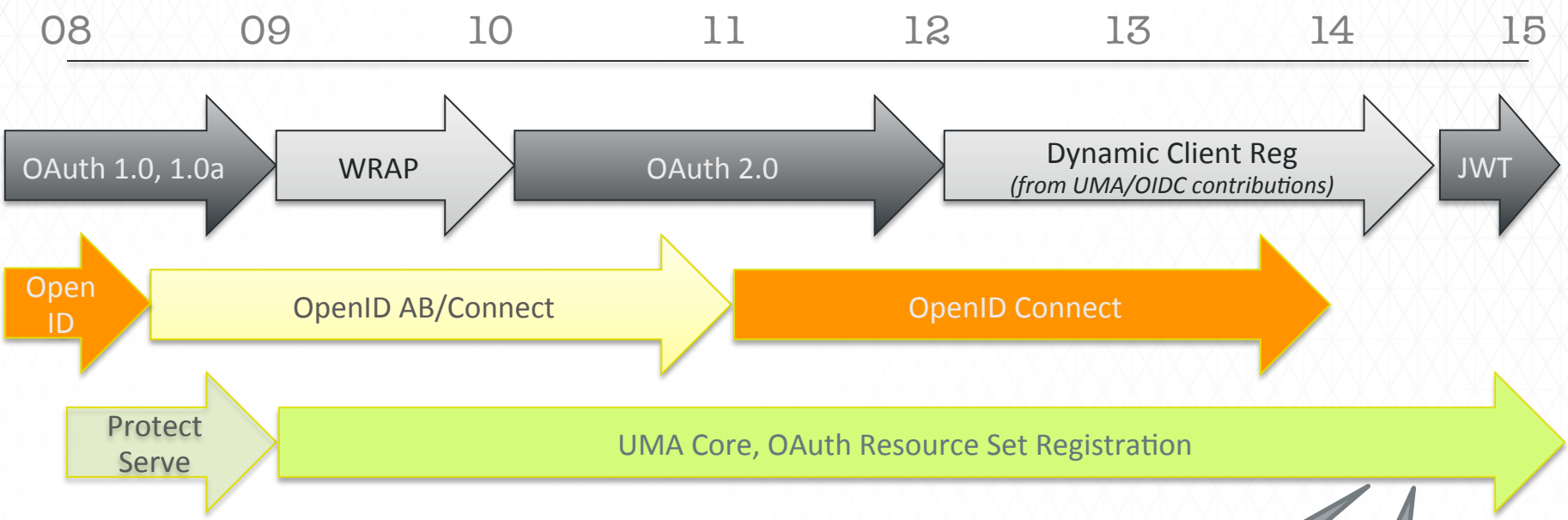
We have tough requirements for delegated authorization

- Lightweight for developers
- Robustly secure
- Privacy-enhancing
- Internet-scalable
- Multi-party
- Enables end-user convenience

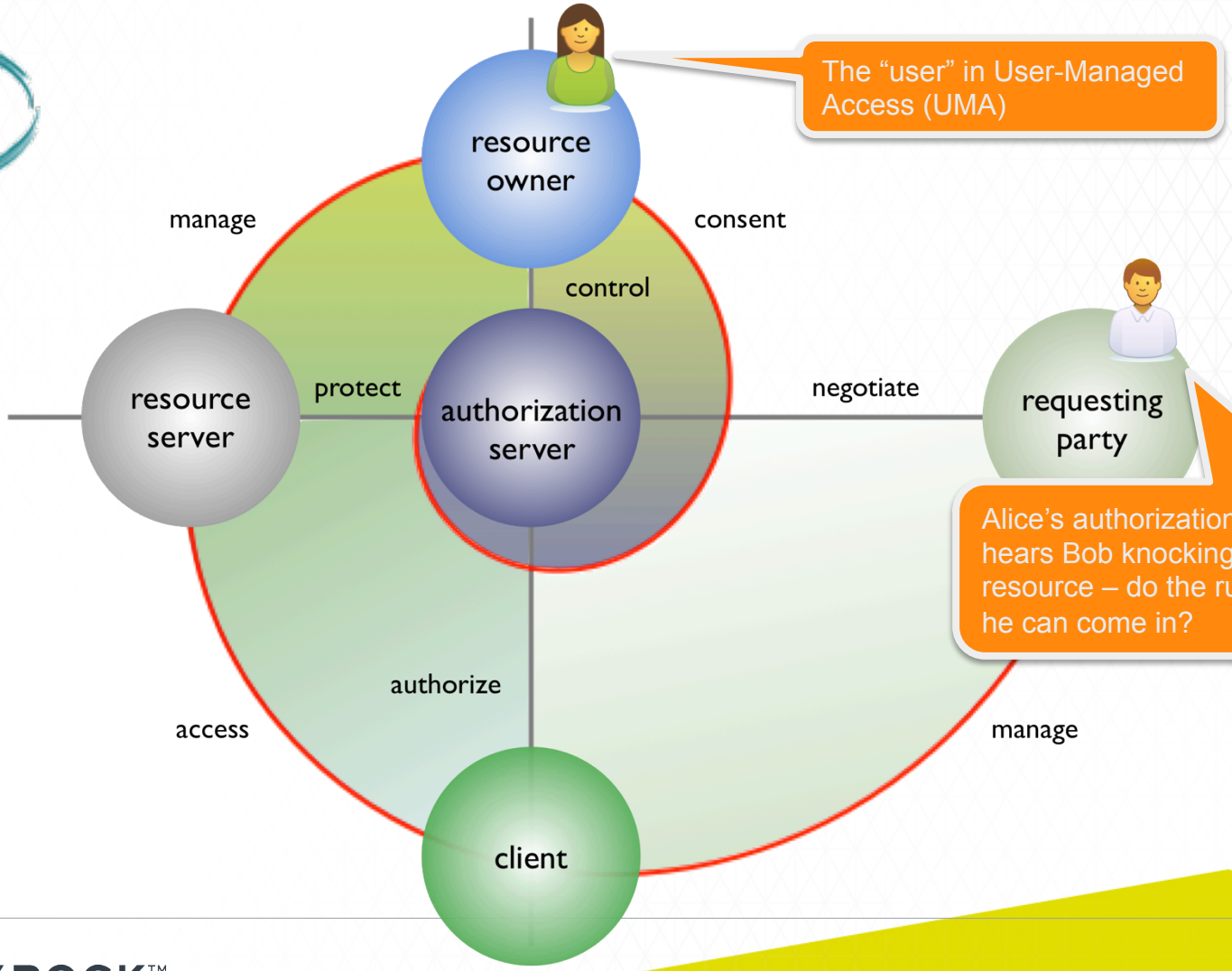
The new Venn of access control



UMA protocol standardization in context



UMA turns online sharing into a Privacy-by-Design solution



The "user" in User-Managed Access (UMA)

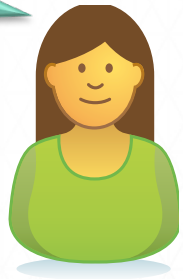
Alice's authorization service hears Bob knocking on the resource – do the rules say he can come in?

UMA-enabled systems can respect policies such as…

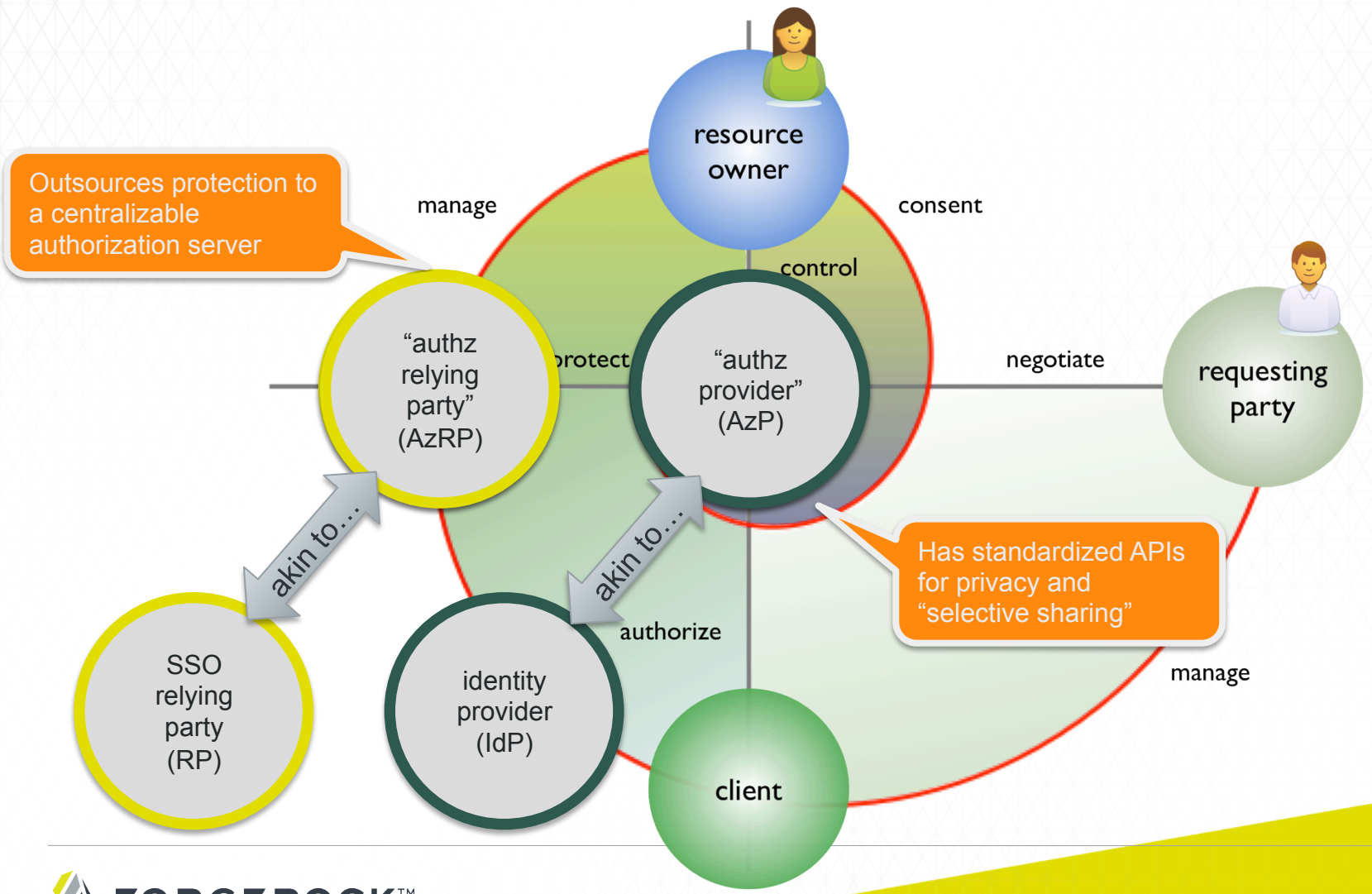
Only let my tax preparer with email TP1234@gmail.com and using client app **TaxThis** access my **bank account data** if they have **authenticated strongly**, and **not after tax season is over**.

Let my **health aggregation app**, my **doctor's office client app**, and the client for my husband's employer's **insurance plan** (which covers me) get access to my **wifi-enabled scale API** and my **fitness wearable API** to **read** the results they generate.

When a person driving a vehicle with an **unknown ID** comes into contact with my **Solar Freakin' Driveway**, alert me and **require my access approval**.



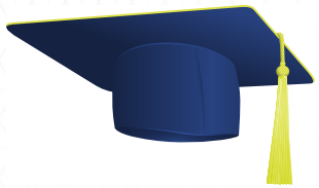
UMA is about interoperable, RESTful authorization-as-a-service



Use-case scenario domains

Health

Financial

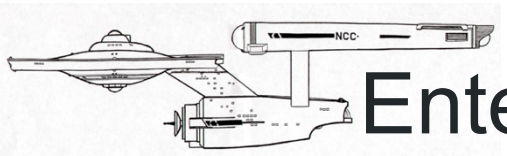


Education

Personal

Government

Media



Enterprise

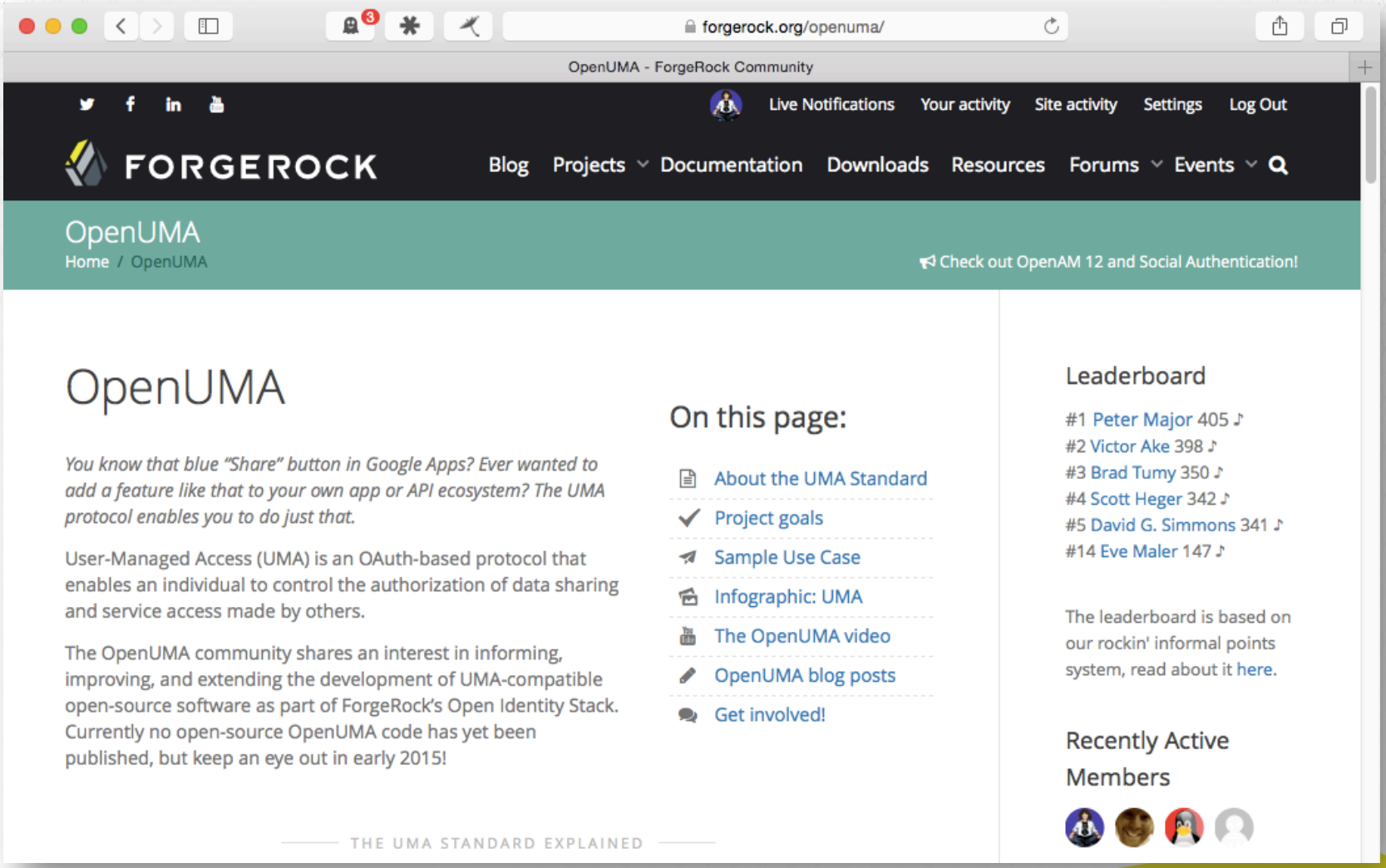
Web

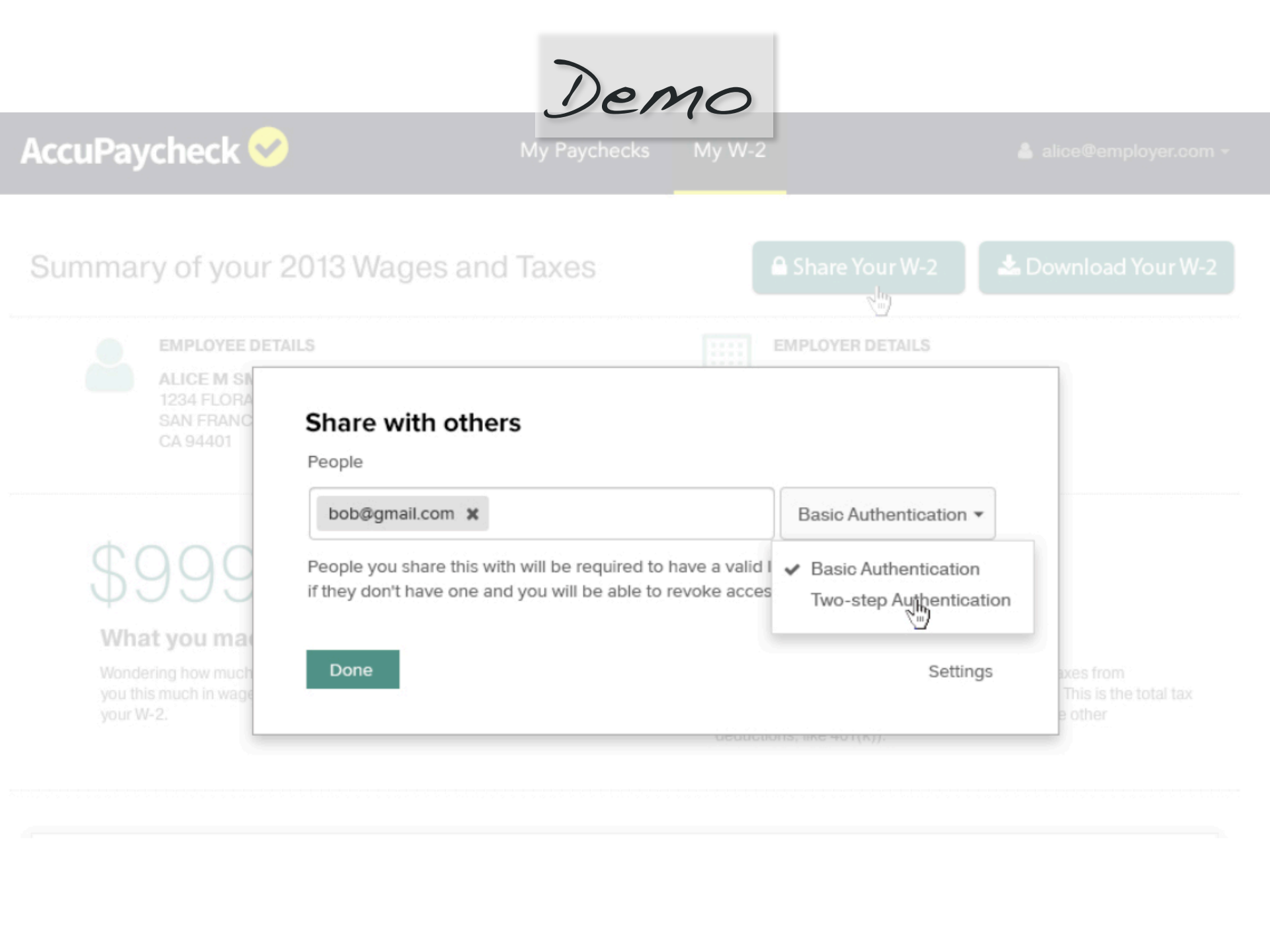
Mobile

API

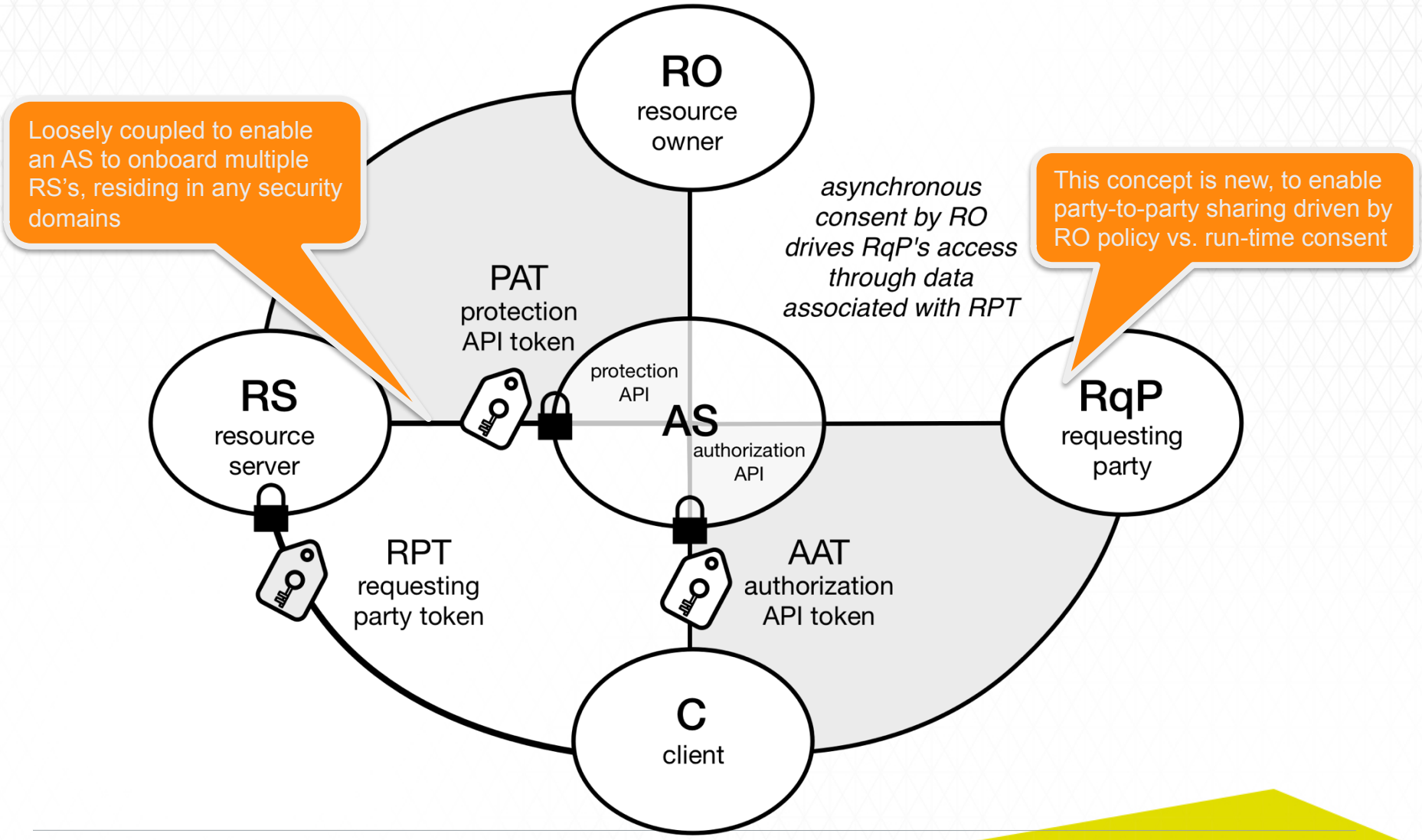
IoT

Introducing the OpenUMA community open-source project

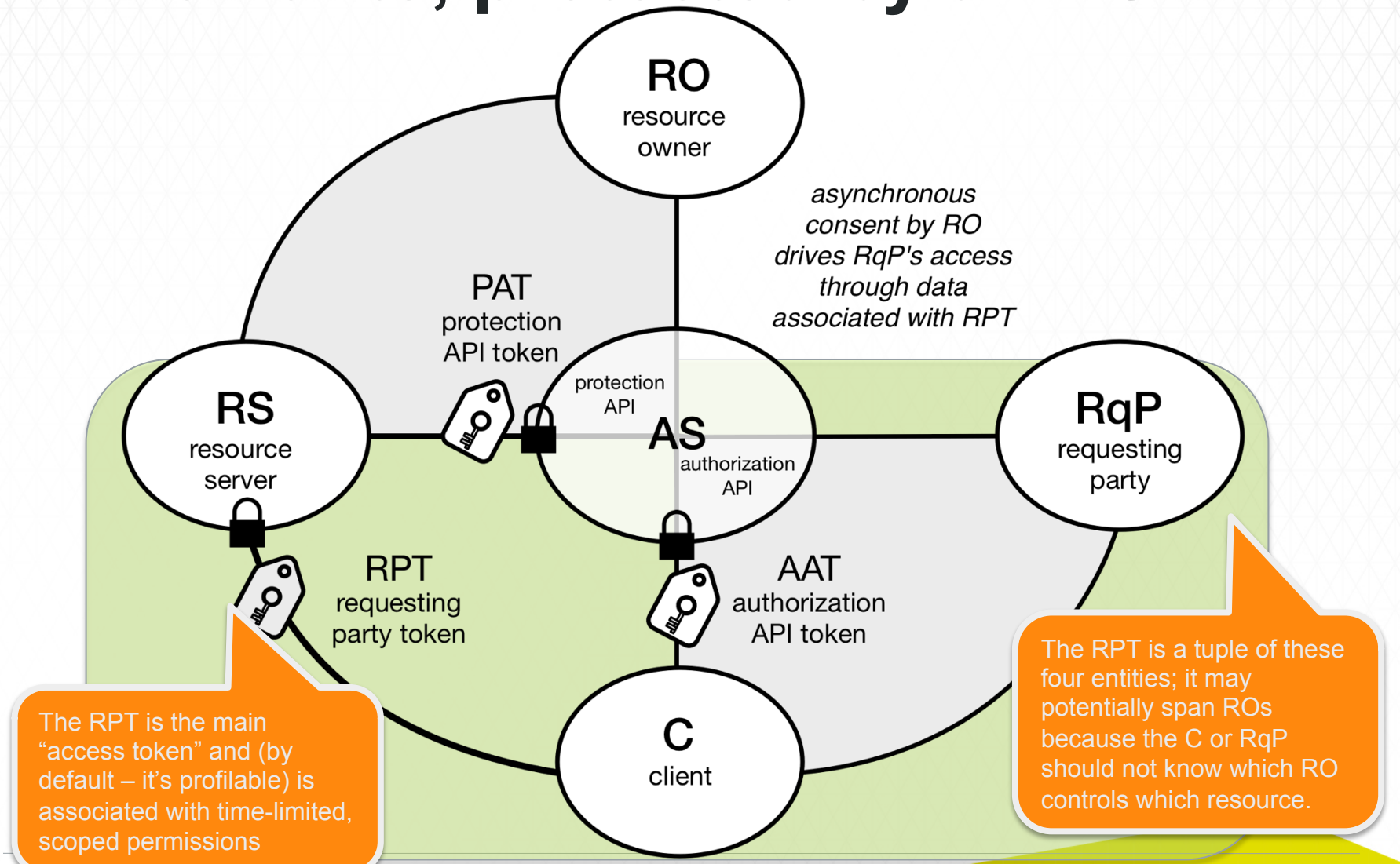



Demo
AccuPaycheck
My Paychecks
My W-2
alice@employer.com
Summary of your 2013 Wages and Taxes
Share Your W-2
Download Your W-2
EMPLOYEE DETAILS
EMPLOYER DETAILS
Share with others
People
bob@gmail.com
Basic Authentication
People you share this with will be required to have a valid
if they don't have one and you will be able to revoke acces
Basic Authentication
Two-step Authentication
Done
Settings

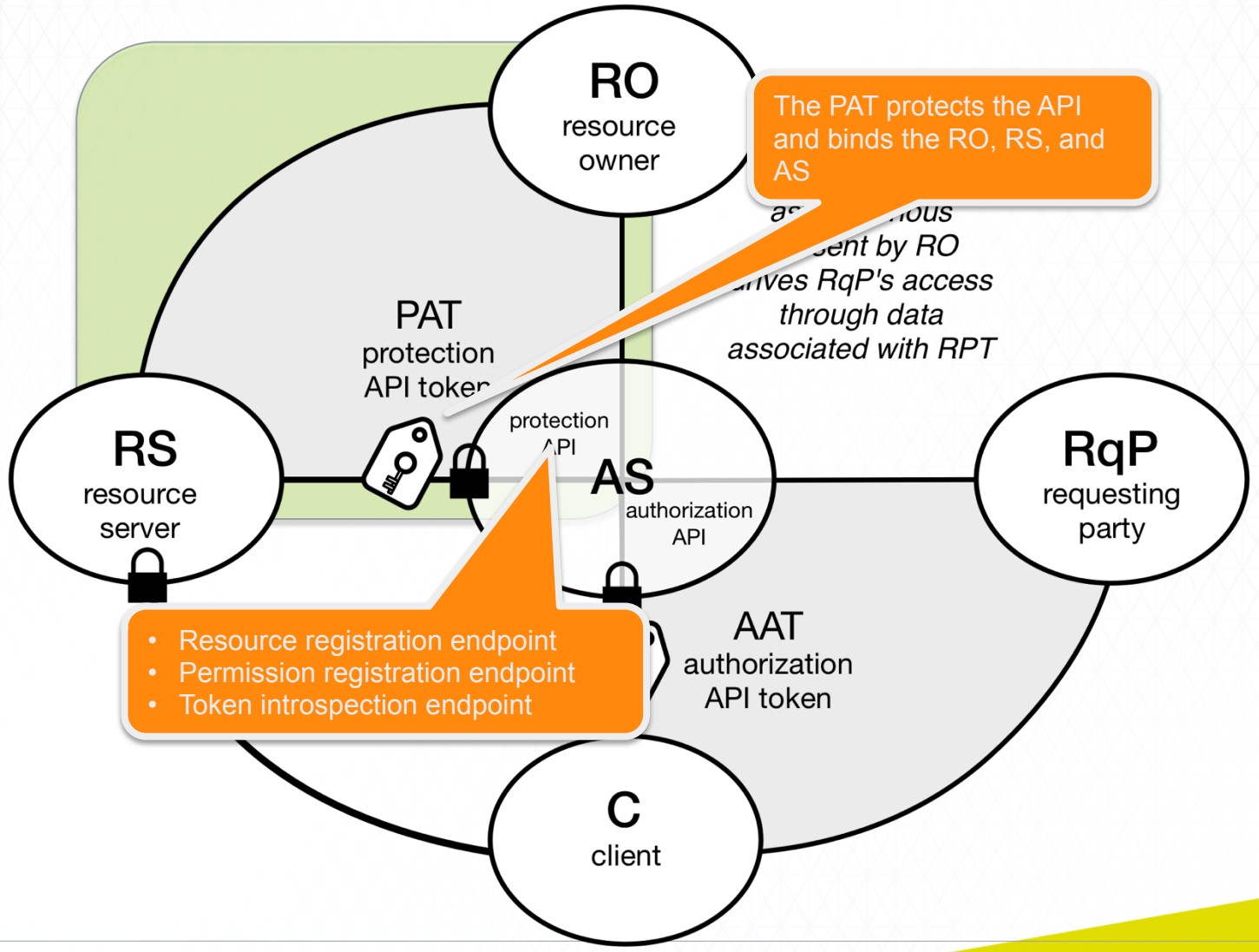
Under the hood, it's "OAuth++"



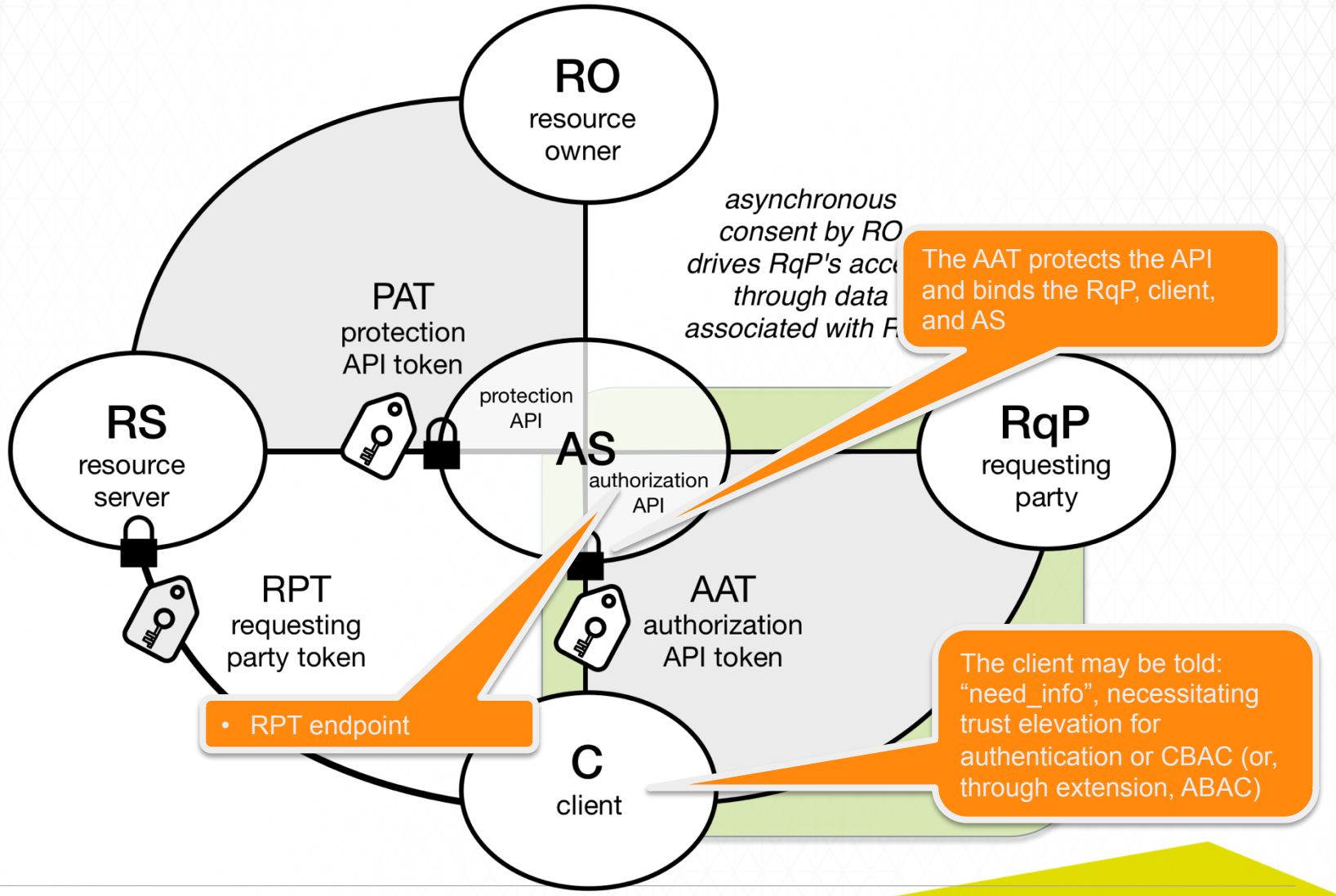
The RS exposes whatever value-add API it wants, protected by an AS



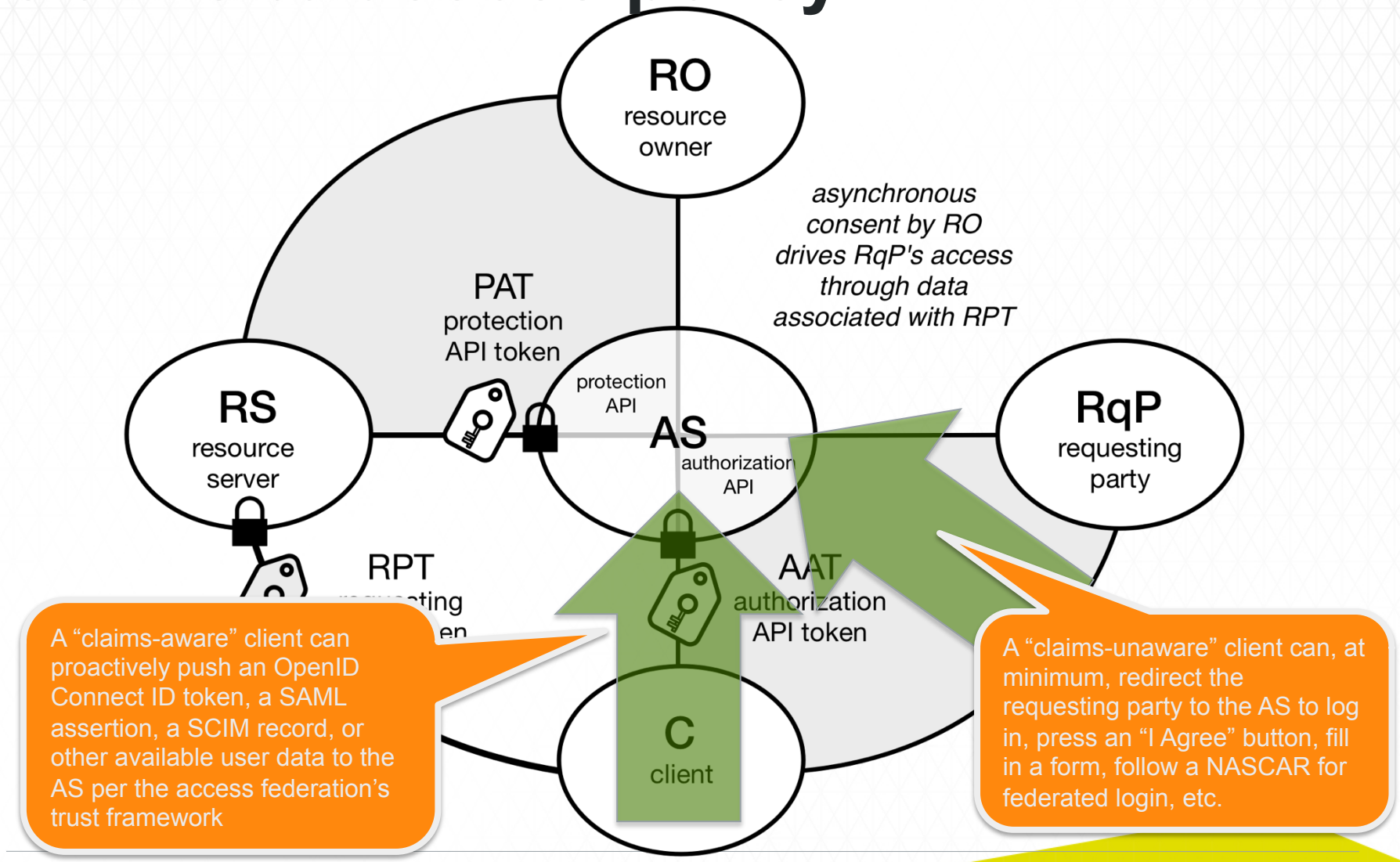
The AS exposes an UMA-standardized protection API to the RS



The AS exposes an UMA-standardized authorization API to the client

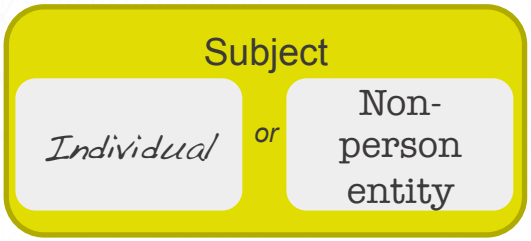


The AS can collect requesting party claims to assess policy



UMA Binding Obligations

- Distributed authorization across domains? Scary!
- This “legal” spec enables parties operating and using software entities (and devices) to distribute rights and obligations fairly in *access federation* trust frameworks



- Authorizing Party
- Requesting Party
- Requesting Party Agent
- Resource Server Operator
- Authorization Server Operator
- Client Operator

Important state changes when new pairwise obligations tend to appear:

- Token issuance
- Token status checks
- Permission registration
- Claims gathering
- Access requests
- Successful access

FORGEROCK™

Thank you!

Eve Maler
VP Innovation & Emerging Technology
eve.maler@forgerock.com
@xmlgrrl